a1 cont

DEVICES," filed on May 28, 1999; U.S. Patent Application No. 09/322,852, entitled "METHOD AND SYSTEM FOR MANAGING SOFTWARE COMPONENTS," filed on May 28, 1999; U.S. Patent Application No. 09/322,965, entitled "METHOD AND SYSTEM FOR IMPLEMENTING VIRTUAL FUNCTIONS OF AN INTERFACE," filed on May 28, 1999; and U. S. Patent Application No. 09/322,457, entitled "METHOD AND SYSTEM FOR PROPERTY NOTIFICATION," filed on May 28, 1999, the disclosures of which are incorporated herein by reference.

Paragraph beginning at line 4 of page 2 has been amended as follows:

a2

A large environment, such as a large building or a large house, may have many audio/video devices located throughout the environment. These AV devices may include CD players, speaker systems, computer systems, television receivers, satellite receivers, displays, and so on. In addition, many sources of media may be available. One such media is a jukebox containing a variety of compact discs. The AV devices typically provide a control panel through which the device can be controlled. For example, a CD player provides a control panel that allows a CD to be started, paused, or stopped. Typically, the interconnections between the AV devices are static. That is, when the AV devices are installed, cabling is routed between devices. For example, speaker wire may be routed between an amplifier and speakers.

Beginning at line 9 of page 3, please insert the following new paragraph:

a3

Figure 9 is a flow diagram illustrating an example implementation of an activation of an AV source port function.

Paragraph beginning at line 6 of page 5 has been amended as follows:

a4

Figure 1 is a block diagram illustrating network layer objects that model the path between output components, switching mechanisms, and input components. In this example, a laserdisc player is connected to a speaker system and a display. The laserdisc player

includes three physical source ports: one for digital video, one for left audio, and one for right audio. The source ports have a direct path to input switching ports of the switching mechanism. The speaker system has two sink ports: one for left audio and one right audio. The display has a sink port for digital video. The sink ports of the output devices have direct paths to the output switching ports of the switching mechanism. The AV system represents each of these components with a corresponding object in memory. The player recorder object 101 corresponds to the laserdisc player. The speaker system object 102 corresponds to the speaker system, and the display object 103 corresponds to the display. The AV system represents multiple ports of a component by a single aggregating port object. The source port object 104 corresponds to the source ports of the laserdisc player, the sink port object 105 corresponds to the sink ports of the speaker system, and the sink port object 106 corresponds to the sink port of the display. Each port object may contain nested port objects to organize the ports of a component in a hierarchy. In this example, the source ports of the laserdisc player are represented by an aggregate source port object 104 that contains two child source port objects. A one child source port object 107 represents the audio source ports, and the other child source port object 108 represents the video source port. The source port object representing the audio source port contains two source port objects. One source object 109 represents the left audio source port, and the other source port object 110 represents at the right audio source port. Similarly, the sink port object 105 represents the sink ports of the speaker system and contains two child sink ports. One sink port object 111 represents the left audio sink port, and the other child sink port object 112 represents the right audio sink port. Since the display has only one sink port, its corresponding sink port object 106 has no child sink ports. A source port object or a sink port object that has no child port is referred to as a primitive port object. For example, source port objects 109 and 110 are primitive source ports. A port object that is not a child of any other port object is referred to as a complete port object. For example, source port object 104 is a complete source port object. Sink port object 106 is both a primitive sink port object and a complete sink port object.

[Paragraph beginning at line 10 of page 6 has been amended as follows:]

The AV system may represent each path by a primitive circuit object. In this example,

primitive circuit object 113 corresponds to a direct path between the left audio source port of the laserdisc player and an input switch port of the switching mechanism. The AV system represents the switching mechanism by a switch object 114. A switch object contains an input source port object 115 for each of its input switch ports and an output switch port object 116 for each of its output switch ports.

-A4 cont

[Paragraph beginning at line 16 of page 6 has been amended as follows:]

The AV system represents a path for a signal between a complete source port and a complete sink port by a virtual circuit. A signal models actual informational context that is on a path. A virtual circuit may represent static and dynamic connections. Figure 2 is a block diagram illustrating the session layer objects that represent virtual circuits. The AV system represents a virtual circuit by a virtual circuit object. The virtual circuit object 201 corresponds to the path between the complete source port of the laserdisc player and the complete sink port of the speaker system. The virtual circuit object 202 corresponds to the path between the source port of the laserdisc player and the complete sink port of the display. The virtual circuit object 201 corresponds only to the audio source ports of the laserdisc player, and the virtual circuit object 202 corresponds only to the video source ports of the laserdisc player. Each virtual circuit object contains a primitive binding information corresponding to each of the paths within that virtual circuit. For example, the virtual circuit object 201 contains a primitive binding information 203 and 204. The AV system allows each source port to be connected to multiple sink ports.

[Paragraph beginning at line 3 of page 7 has been amended as follows:]

Figure 3 is a block diagram illustrating management layer objects. The AV system represents the signals that are output by the source ports of an output component as a stream. That is, each output outputs a stream of signals. The signals within the stream are hierarchically organized in a manner that is similar to how source ports are organized within a complete source port. The AV system represents the stream of an output component by a stream object that may contain other stream objects. In this example, the output signals of the

laserdisc player are represented by stream object 301. The audio signals of the laserdisc player are represented by child stream object 302, and the video signal of the laserdisc player is represented by child stream object 303. The audio stream object contains a child stream object 304 representing the left audio signal and a child stream object 305 representing the right audio signal. A stream object that does not contain other stream objects is referred to as a primitive stream object. A stream object that is not contained within other stream objects is referred to as a complete stream object. For example, stream object 301 is a complete stream object, and stream object 304 is a primitive stream object. Each primitive stream object contains a signal object that corresponds to the signal that is output by the corresponding source port. Signal object 306 corresponds to the signal that is transmitted between the left audio source port of the laserdisc player and the left sink port of the speaker system. Signal object 307 corresponds to the signal that is transmitted between the right audio source of the laserdisc player and the right sink port of the speaker system. Signal object 308 corresponds to the signal that is transmitted from the video source port of the laserdisc player to the sink port of the display.

Paragraph beginning at line 9 of page 10 has been amended as follows:

This function returns an indication as to whether this port is a complete port.

Paragraph beginning at line 12 of page 10 has been amended as follows:

This function returns an indication as to whether this port is a primitive port.

Paragraph beginning at line 15 of page 11 has been amended as follows:

This function informs a sink port that it is to consider the signals within a stream for the purpose of assigning them to a primitive sink port.

Paragraph beginning at line 19 of page 13 has been amended as follows:

Q8

This function returns an indication as to whether this stream is a primitive stream.

Paragraph beginning at line 9 of page 14 has been amended as follows:

Q9

This function returns a pointer to the signal in this stream as a primitive stream.

Paragraph beginning at line 2 of page 15 has been amended as follows:

Q10

This function returns the number of connections from this input switch port to output switch ports.

Paragraph beginning at line 13 of page 16 has been amended as follows:

Q11

This function returns the number of bindings between primitive source ports and primitive sink ports for this virtual connection.

Paragraph beginning at line 1 of page 17 has been amended as follows:

Q12

This function returns an indication of whether this source is active. A source port is active when it is capable of producing a signal.

Paragraph beginning at line 8 of page 17 has been amended as follows:

Q13

This function returns a pointer to the primitive circuit associated with this source port.

Paragraph beginning at line 25 of page 17 has been amended as follows:

Figure 10 is a block diagram illustrating the components of an entertainment center. An entertainment center component provides a behavior that allows an AV program to be assigned to a player/recorder component. When a program is assigned to an entertainment center, the entertainment center performs the processing that is needed to load that program into a player/recorder, cause the program to be played, and route the output signals of the player/recorder component to output components. An entertainment center may be associated with a space (e.g, a room within a house). The entertainment center may also be associated with multiple player/recorders and multiple output components such as a display component and a speaker subsystem component. The AV system represents the associated space by a space object 1001, represents the player/recorder components by player/recorder objects 1002, and represents the output components by a display object 1003 and a speaker subsystem object 1004. An entertainment center may have a default set of the output components. When a program is assigned to the entertainment center, the output signals for the player/recorder component are routed to these default output components. The entertainment center controls the creating of virtual circuits that are needed to effect this routing. The entertainment center may also allow the output signals of a player/recorder component to be dynamically routed to different output components. For example, the entertainment center may allow the output of the player/recorder component to be dynamically routed to a speaker system component associated with another space. To effect this dynamic routing, the AV system creates and destroys virtual circuits dynamically. In one embodiment, the entertainment center may determine for each of its output components whether the routing should be allowed, whether to be notified when an output signal is routed due to an action external to the entertainment center, and whether to provide a user interface for controlling the output component to which the signal is routed. These determinations may be different for each output component associated with the entertainment center. When an entertainment center is notified that one of its output components has been routed due to an external action (e. g. a different entertainment center routing to the output component causing the notification), the entertainment center can become an additional controller of the player/recorder. An entertainment center may also provide property notifications when the properties of its

A14 cont

associated player/recorder components or output components change. For example, the entertainment center may notify a corresponding user interface component that the pause button on a player/recorder component has been depressed. An entertainment center object may provide a user interface component that is appropriate for controlling the user interface of the input components and output components associated with the entertainment center.

Paragraph beginning at line 3 of page 20 has been amended as follows:

A15

A program pool object represents a collection of AV programs. Each AV program has a corresponding program object. An AV program conceptually corresponds to a media that can be played by a player/recorder component. For example, an AV program may represent the feed provided through a certain television channel, a musical score stored on a CD, a movie stored on a laserdisc, and so on. These AV programs can be hierarchically organized to represent more complex AV programs. For example, an AV program may include a sub-AV program corresponding to the feed from a television channel and a sub-AV program corresponding to the output of a computer program. Thus, AV programs can represent arbitrarily complex multimedia programs. The AV system represents an AV program by a program object. A program object provides the behavior to browse through the hierarchy of the AV programs represented by that program object, allows a player/recorder component to be assigned to the AV program, and provides a behavior corresponding to the loading of the AV program into the player/recorder component. A program object also has a program ID, which provides descriptive information about the AV program. For example, descriptive information may include the name of the movie that the AV program represents. A program object stores the location of the media that corresponds to the AV program. For example, if the AV program corresponds to a laserdisc in a certain laserdisc stack, then the location would indicate the stack and slot of the laserdisc within the stack. In one embodiment, the location is represented as a path within a hierarchy of locations. A program object stores the identifier of an owner, which may be the program pool object that the program object is within. A program object allows for the retrieving of its child program objects and may allow for certain criteria to be established so that only children that match the criteria are returned. A program object may also allow for retrieving of its parent program object. In one

embodiment, the parent program object may be retrieved through the containing program pool by providing the location of the program object to the program pool. A program object has a program type associated with it. The program type specifies a path through a hierarchy of program types. The hierarchy of program types is described below in detail.

Paragraph beginning at line 18 of page 21 has been amended as follows:

A program pool has a corresponding program pool object. A program pool object provides an access port for each client that is accessing the program pool. The program pool object provides a function that receives a program ID and returns a reference to a program object corresponding to that program ID. A program pool object also allows for database cursor-like access to the program objects. For example, a query can be submitted which specifies the criteria for program objects. The program objects that match that criteria are provided in a result set. The client can access that result set using techniques such as advance to the next program object, get reference for the current program object, and return a set of references for the program objects in the result set. In one embodiment, the result set of a query may be cached at a client to reduce communications between the clients in the program pool. The program pool may also automatically update the client's cache as the set of programs that match the criteria changes. In one embodiment, the program pool provides an access control mechanism to restrict access by certain clients. The program pool may use the phantom object mechanism as described in commonly assigned copending U.S. Patent Application No. 09/322,455, entitled "Method and System for Tracking Clients," filed May 28, 1999.

Paragraph beginning at line 20 of page 22 has been amended as follows:

Figure 12 is a flow diagram illustrating the assigning of a program to an entertainment center. In step 1201, the function invokes a function to select a certain program object. The invoked function returns a pointer to the program object. In step 1202, the function invokes the set current program function of the entertainment center object passing the pointer to the program object. The processing is then complete.

Paragraph beginning at line 25 of page 22 has been amended as follows:

Figure 13 is a flow diagram of a function to select a program. This function may display a user interface that allows a user to browse through the programs in a program pool. The user interface may allow the user to specify various search criteria. For example, the user interface may allow the user to specify the type of music that is of interest. In step 1301, the function allows the user to select a program from the program pool. In step 1302, the function sets the return pointer to a pointer to a program object representing the program. The function then returns.

Paragraph beginning at line 6 of page 23 has been amended as follows:

Figure 14 is a flow diagram representing an example implementation of a set of current program function of an entertainment center object. This function is passed a pointer to a program object and effects the loading of that program within the entertainment center. In step 1401, the function invokes a function to retrieve a loaded player/recorder object. The function passes a pointer to the program object and is returned a pointer to a player/recorder object that is loaded with the program. In step 1402, the function invokes the get current source function of the player/recorder object. That invoked function returns a pointer to the complete source port for the player/recorder object. In step 1403, the function invokes the get stream pointer function of the source port object to retrieve a pointer to the complete stream for that source port object. In steps 1404-1407, the function loops selecting the output components associated with the entertainment center and creating a virtual circuit from the player/recorder component to the output components. As described above, a entertainment center may have a default set of output components. In step 1404, the function selects the next output component. In step 1405, if all the output components have already been selected, then the function returns, else the function continues at step 1406. In step 1406, the function requests the selected output component to return a sink port object that is appropriate to the stream. The function invokes a get sink port function of the output object corresponding to the selected output component. In step 1407, the function invokes the create virtual circuit function of the source port object passing a pointer to the sink port object. That invoked

A17 cont

function creates a virtual circuit from the source port to the sink port. The function then loops to step 1404 to select the next output component.

Paragraph beginning at line 3 of page 25 has been amended as follows:

Figure 18 is a flow diagram of an example load program function of a media manager object. This example function describes the processing that may be performed when the media manager has child media manager objects. This function is passed a pointer to a program object and returns a pointer to a player/recorder object. In step 1801, the function invokes the get location function of the program object to retrieve the location of the media as indicated by the program object. In step 1802, the function searches the location table for a media manager object that manages the media corresponding to the program object. In step 1803, the function invokes the load program function of the located media manager object and then returns.

A18

Paragraph beginning at line 34 of page 26 has been amended as follows:

One skilled in the art would appreciate that various modifications can be made to the present invention. Accordingly, the invention is not limited to the specific embodiments, but instead the scope of an invention is specified by the following claims.

A19

IN THE CLAIMS:

Please amend claims 1-3 as follows:

1. An audio/visual system, comprising:

at least one output component having at least one source port for each type of output signal output from the at least one output component and at least one source port object for each of said at least one source port; and

A20

at least one input component having at least one sink port for each type of input signal input to the at least one input component and at least one sink port object for each at least one sink port,